XML Vignette

Shaleni Kovach

10/13/2019

Data Type - XML

This vignette covers XML data. JSON stands for Extensible Markup Language, and is commonly used to transfer an display data, particulary in website settings - though JSON has been gaining popularity over it for transportation/storage purposes due to differences in computation speed between the two.

XML has two main components - markup and content. Markup tags are used to specify what content is/how it should be displayed, while content is the actual information to be portrayed.

Reading in XML files

The main package for reading XML data in R is the XML package. It has an overwhemling presence when searching for XML packages in R.

Dataset

This example will use data from a XML Rabies Lab Submissions from New York (website link here).

This dataset contains counts of different animals which were tested for rabies, and if they came back positive or not for various counties in New York.

The data contains 123 rows and 24 columns, where each row is a county.

EDA

Reading in the data:

```
dataXML <- xmlParse(file = "rabies.xml")
rootNode <- xmlRoot(dataXML)
dataXML <- xmlSApply(rootNode,function(x) xmlSApply(x, xmlValue))

data <- fromJSON("https://health.data.ny.gov/resource/56p2-pcd9.json")


data$year <- as.factor(data$year)
data$countyname <- as.factor(data$countyname)
data <- data %>% mutate_if(is.character, as.numeric)

head(data)
```

```
##   year  countyname cattle cattlepos bat batpos cat catpos fox foxpos dog
## 1 2018      Albany      1         0 111      3  36      1   1      1  20
## 2 2018    Allegany      0         0  21      2   4      0   1      1   3
## 3 2018       Bronx      0         0   2      0   2      0   0      0   1
## 4 2018      Broome      0         0  36      0  24      1   1      0  12
## 5 2018 Cattaraugus      2         0  68      5   6      0   3      0   4
## 6 2018      Cayuga      0         0  61      0   7      0   2      1  12
```

```
##   dogpos raccoon raccoonpos skunk skunkpos domother domotherpos wildother
## 1      0      10          5     4        1        2           0         6
## 2      0       4          0     0        0        1           0         5
## 3      0       1          0     1        1        0           0         0
## 4      0       1          1     0        0        2           0        10
## 5      0      11          2     0        0        3           0        11
## 6      0       8          7     3        2        3           0         4
##   wildotherpos rodents rodentspos sent totalpos
## 1            0      12          0  203       11
## 2            1       1          0   40        4
## 3            0       1          0    8        1
## 4            0       1          0   87        2
## 5            0       2          1  110        8
## 6            0       2          0  102       10
```

The data could use some extra variables - a fraction of the rabies positive animals by species, for instance.

```
generateFraction <- function(data=data, name) {
  total <- data[[name]]
  total <- as.numeric(total)
  posname <- paste0(name,'pos')
  positive <- data[[posname]]
  positive <- as.numeric(positive)
  fracname <- paste0(name,'fraction')
  return(positive/total)
}

names <- c("cattle", "bat", "cat", "fox", "dog", "raccoon", "skunk", "domother", "wildother","rodents")
fracnames <- character()
for (name in names){
  newcol <- paste0(name,"fraction")
  fracnames <- c(fracnames,newcol)
  data[[newcol]] <- numeric(length(data[[name]]))
  data[[newcol]] <- generateFraction(data,name)
}
```

Now that we have the data read in and some additional variables, we can take a look at some of the numbers.

```
summary(data$totalpos)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    3.00   14.00   53.29   61.50  929.00
```

```
summary(data$sent)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     3.0    14.0    53.2    56.5   894.0
```

Hmm. That doesn't look right. The maximum number of animals that tested positive for rabies shouldn't be greater than the maximum number of animals sent in... Let's look at that a little more closely.

```
head(data %>% arrange(desc(totalpos)) %>% select(year,countyname,sent,totalpos))
```

```
##   year  countyname sent totalpos
## 1 2017        Erie   15      929
## 2 2017    Onondaga   17      425
## 3 2017 Westchester   18      414
## 4 2017      Nassau    2      247
```

```
## 5 2017      Albany   13      221
## 6 2017    Tompkins   13      218
```

Yep that definately does not look right. Let's remove the rows that have greater total positive than sent.

```
data <- data %>% filter(sent>totalpos)
data <- data %>% mutate(totalfrac = totalpos/sent)

summary(data$totalpos)
```

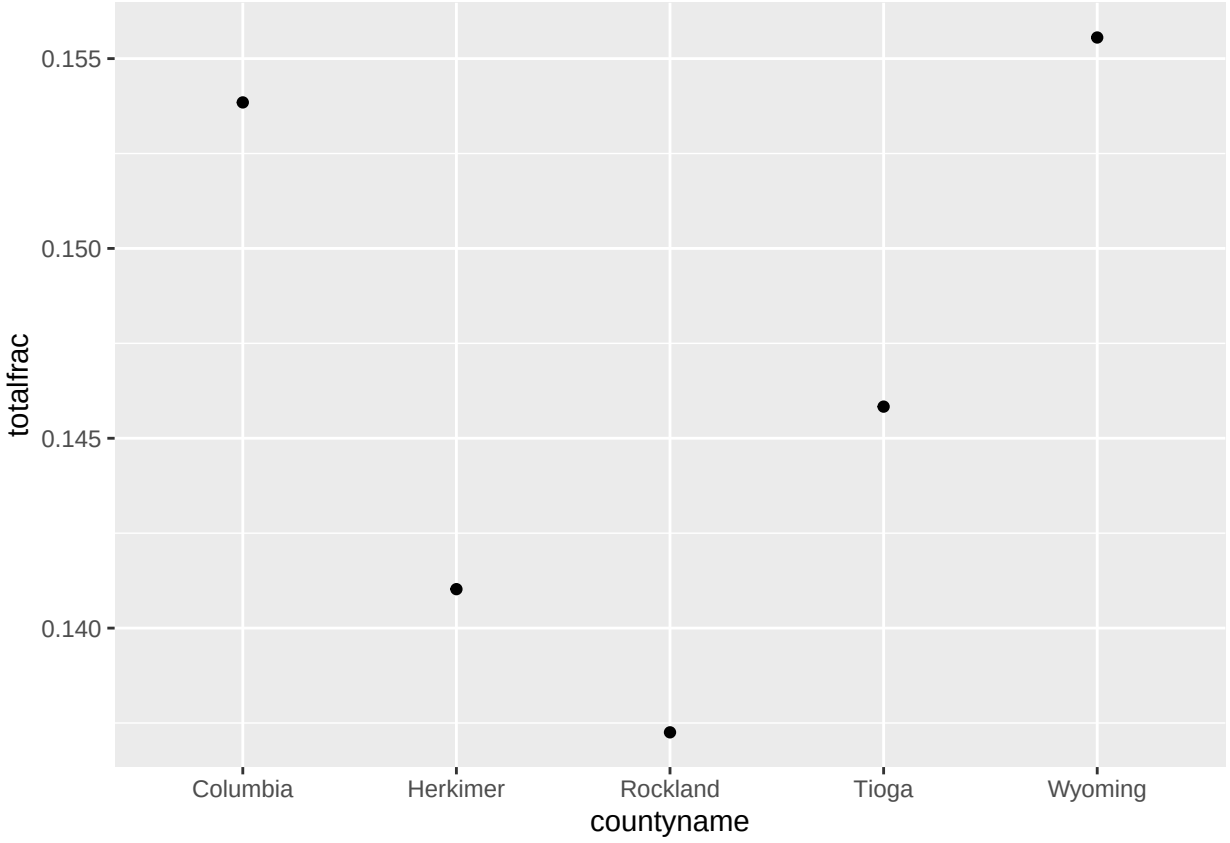
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   1.000   3.000   5.194   7.750  22.000
```

```
summary(data$sent)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     1.0    37.0    56.5   100.9   110.8   894.0
```

That looks much more reasonable. Now let's take a look at some graphs.

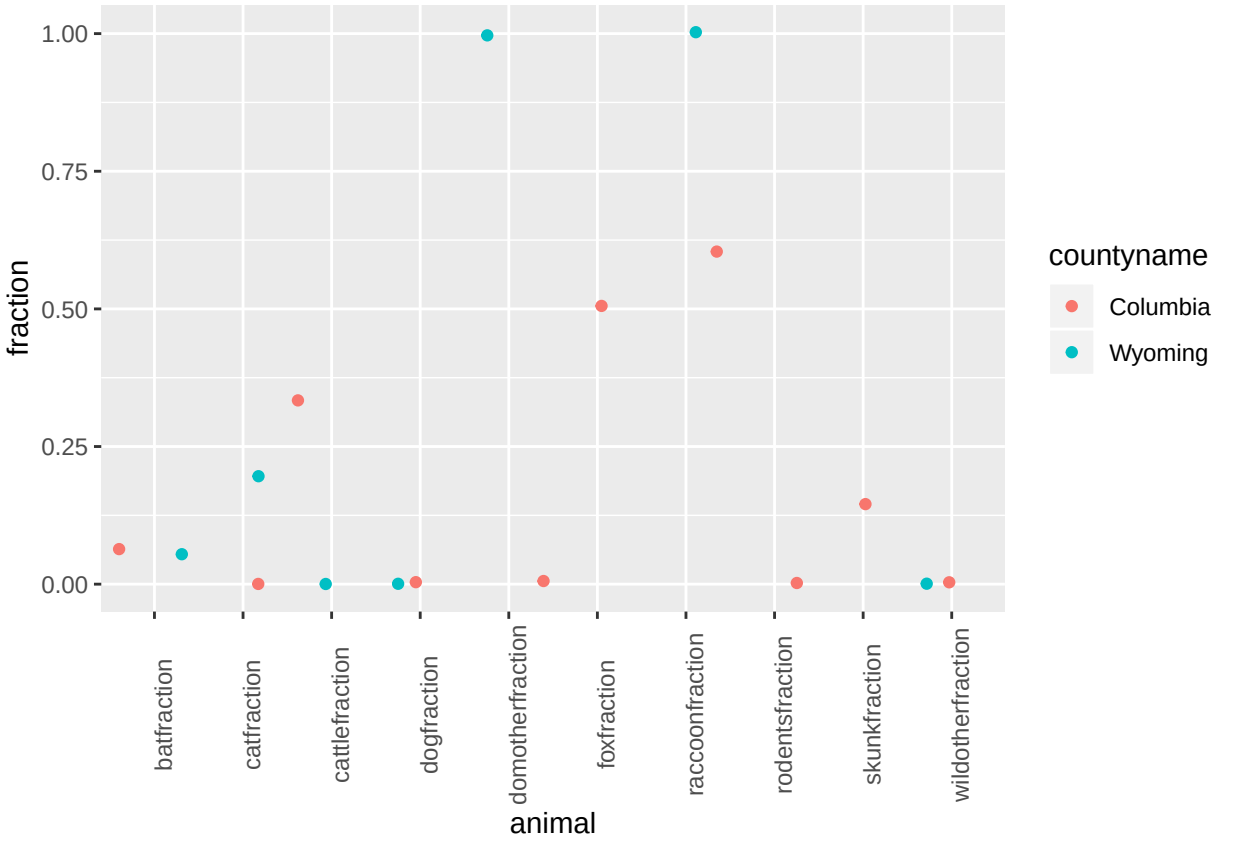
```
topFrac <- data %>% arrange(desc(totalfrac))
ggplot(topFrac[1:5,], aes(countyname,totalfrac)) + geom_point()
```



It looks like Wyoming and Columbia have the highest overall fraction of rabies in their animals.

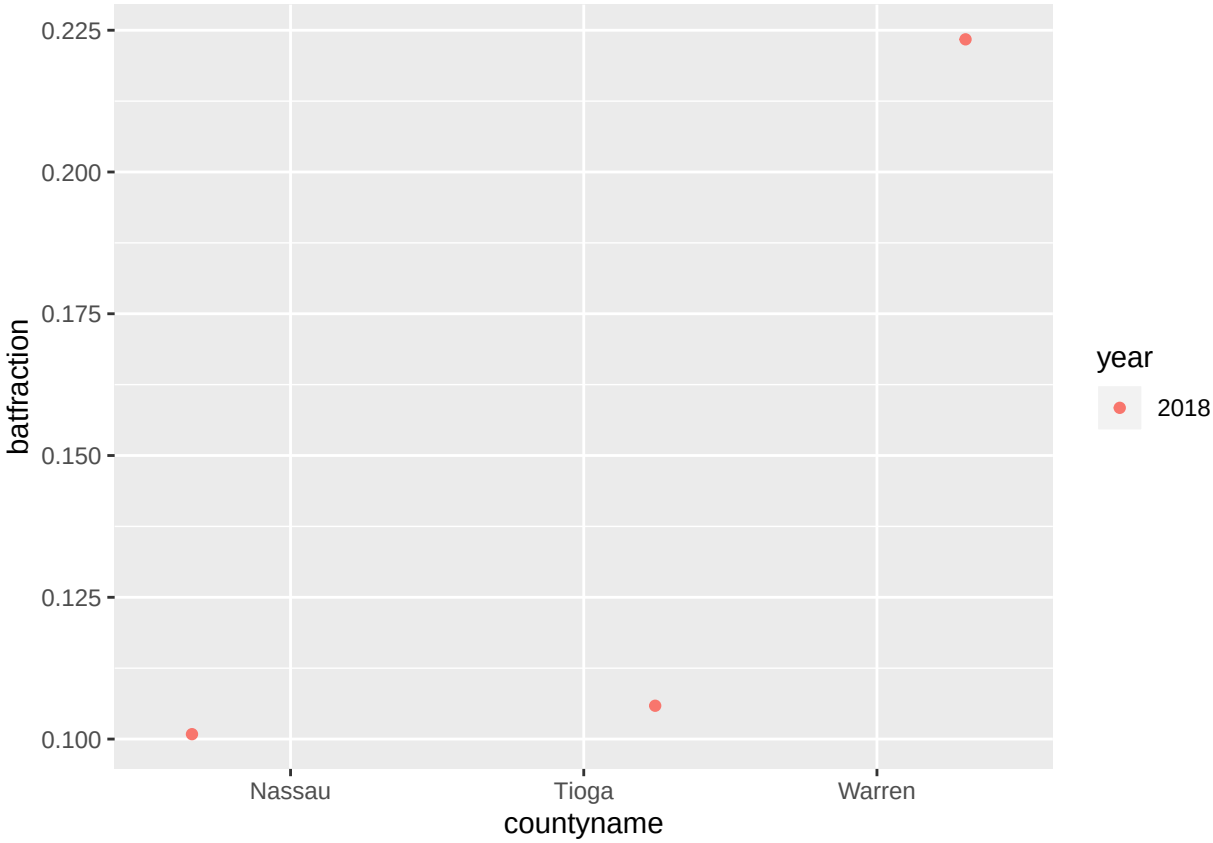
Let's look at Wyoming and Columbia a little more in-depth.

```
ggplot(highest,aes(animal,fraction,color=countyname)) + geom_jitter() + theme(
    axis.text.x=element_text(angle=90))
```



100% of Wyoming's domestic other animals and raccoons came back with rabies. Don't want to come across either of those running around there! Columbia's results were overall more moderate, with a little over 50% of those animals coming back positive with rabies.

```
bats <- data %>% arrange(desc(batfraction)) %>% filter(batfraction >= .1)
ggplot(bats,aes(countyname,batfraction,color=year)) + geom_jitter()
```


0.225
0.200
0.175
0.150
0.125
0.100
batfraction
Nassau
Tioga
Warren
countyname
year
2018